Algorithms that learn to think on their feet
(now, with amazing bonus prize!)

Hal Daumé III | University of Maryland | me@hal3.name | @haldaume3
What is NLP?

- Fundamental goal: deep understanding of text
  - Not just string processing or keyword matching
- End systems that we want to build
  - Simple: Spelling correction, text categorization, etc.
  - Complex: Speech recognition, machine translation, information extraction, dialog interfaces, question answering
  - Unknown: human-level comprehension (more than just NLP?)
Why is language hard?

- Ambiguity abounds (some headlines)
  - Iraqi Head Seeks Arms
  - Teacher Strikes Idle Kids
  - Kids Make Nutritious Snacks
  - Stolen Painting Found by Tree
  - Local HS Dropouts Cut in Half
  - Enraged Cow Injures Farmer with Ax
  - Hospitals are Sued by 7 Foot Doctors
  - Ban on Nude Dancing on Governor's Desk
  - Scientists study whales from space

- Why are these funny?

- What does ambiguity imply about the role of learning?
Despite ambiguity, language is predictable

I like my coffee with cream and **asparagus**

This is crummy weather for San **ta Claus**

- The brain uses this information!

- Can we use predictability to make decisions *before* all of the input is observed?

YES!!!
Outline

Quizbowl (Incremental Question Answering)
Simultaneous (machine) interpretation

➢ Dozens of defendants
➢ Judges from four nations (three languages)
➢ Status quo: speak, then translate
➢ After Nuremberg, simultaneous translations became the norm
➢ Long wait → bad conversation
Why simultaneous interpretation is hard

- Human languages have vastly different word orders
  - About half are OV, the other half are VO
  - This comes with a lot more baggage than just verb-final

Running (German/English) Example:

Ich bin mit dem Zug nach Ulm gefahren
I am with the train to Ulm traveled
I (...... waiting. ......) traveled by train to Ulm
Model for interpretation decisions

- We have a set of actions (predict / translate)
  - Wait
  - Predict clause-verb
  - Predict next word
  - Commit ("speak")

- In a changing environment (state)
  - The words we've seen so far
  - Our models' internal predictions

- With a well defined notion of "optimal action" at training time
Example of interpretation trajectory

1. Mit dem Zug

Verb: **gewesen**

Next: **und**

Ich bin mit dem Zug nach Ulm gefahren
I am with the train to Ulm traveled
I (…… waiting……) traveled by train to Ulm
DAgger: Dataset Aggregation

- Collect trajectories from expert \( \pi^* \)
- Dataset \( D_0 = \{ (s, \pi^*(s)) \mid s \sim \pi^* \} \)
- Train \( \pi_1 \) on \( D_0 \)
- Collect new trajectories from \( \pi_1 \)
  - But let the expert steer!
- Dataset \( D_1 = \{ (s, \pi^*(s)) \mid s \sim \pi_1 \} \)
- Train \( \pi_2 \) on \( D_0 \cup D_1 \)

In general:
- \( D_n = \{ (s, \pi^*(s)) \mid s \sim \pi_n \} \)
- Train \( \pi_n \) on \( \bigcup_{i<n} D_i \)

If \( N = T \log T \),
\[
L(\pi_n) < T \epsilon_N + O(1)
\]
for some \( n \)
Evaluating performance and baselines

![Diagram showing different types of translation performance](image-url)
Evaluating performance and baselines

Grissom II et al., EMNLP 2014
Evaluating performance and baselines
Evaluating performance and baselines

Source Sentence

Psychic

Monotone

Batch

Policy Prediction

He went to the store

He to the

He went

Good Translation

Bad Translation

Good Translation

Bad Translation

Good Translation

Bad Translation

(Grissom II et al., EMNLP 2014)
Evaluating performance and baselines

Grisom II et al., EMNLP 2014
Evaluating performance and baselines

Grissom II et al., EMNLP 2014

Hal Daumé III (me@hal3.name)

Algorithms that think on their feet
Evaluating performance and baselines

(Grisson II et al., EMNLP 2014)
Training the policy

- **Actions:**
  - Commit: `translate(revealed words)`
  - Predict (verb/next): `translate(revealed + predicted)`
  - Wait: `get_next_words()`

- **Delayed feedback:** latency BLEU

- **Features:**
  - Output & confidence of predictors
  - Internal translation / language model scores
  - Previous decisions made by policy
Evaluating performance

- **Batch**
- **Monotone**
- **Optimal**
- **Learned**
Algorithms that think on their feet

Simultaneous Machine Interpretation

Ich bin mit dem Zug nach Ulm gereist

I ( . . . . . . waiting . . . . . )

traveled by train to Ulm

Quizbowl
(Incremental Question Answering)

Mohit Iyyer

He He

Outline
Humans doing incremental prediction

- Game called “quiz bowl”
- Two teams play each other
  - Moderator reads a question
  - When a team knows the answer, they buzz in
  - If right, they get points; otherwise, rest of the question is read to the other team
- Hundreds of teams in the US alone
- Example . . .
Quizbowl example

With Leo Szilard, he invented a doubly-eponymous
Quizbowl example

With Leo Szilard, he invented a doubly-eponymous refrigerator with no moving parts. He did not take interaction with neighbors into account when formulating his theory.
Quizbowl example

With Leo Szilard, he invented a doubly-eponymous refrigerator with no moving parts. He did not take interaction with neighbors into account when formulating his theory of heat capacity, so
Quizbowl example

With Leo Szilard, he invented a doubly-eponymous refrigerator with no moving parts. He did not take interaction with neighbors into account when formulating his theory of heat capacity, so Debye adjusted the theory for low temperatures. His summation convention automatically sums repeated indices in tensor products. His name is attached to the A and B coefficients.
Quizbowl example

With Leo Szilard, he invented a doubly-eponymous refrigerator with no moving parts. He did not take interaction with neighbors into account when formulating his theory of heat capacity, so Debye adjusted the theory for low temperatures. His summation convention automatically sums repeated indices in tensor products. His name is attached to the A and B coefficients for spontaneous and stimulated emission, the subject of one of his multiple groundbreaking 1905 papers. He further developed the model of statistics sent to him by...
Quizbowl example

With Leo Szilard, he invented a doubly-eponymous refrigerator with no moving parts. He did not take interaction with neighbors into account when formulating his theory of heat capacity, so Debye adjusted the theory for low temperatures. His summation convention automatically sums repeated indices in tensor products. His name is attached to the A and B coefficients for spontaneous and stimulated emission, the subject of one of his multiple groundbreaking 1905 papers. He further developed the model of statistics sent to him by Bose to describe particles with integer spin. For 10 points, who is this German physicist best known for formulating
With Leo Szilard, he invented a doubly-eponymous refrigerator with no moving parts. He did not take interaction with neighbors into account when formulating his theory of heat capacity, so Debye adjusted the theory for low temperatures. His summation convention automatically sums repeated indices in tensor products. His name is attached to the A and B coefficients for spontaneous and stimulated emission, the subject of one of his multiple groundbreaking 1905 papers. He further developed the model of statistics sent to him by Bose to describe particles with integer spin. For 10 points, who is this German physicist best known for formulating the special and general theories of relativity?
Solving incrementally

- **Action:** buzz now or wait
  - Content Model is constantly generating guesses
  - Oracle provides examples where it is correct
  - The Policy generalizes to test data
  - Features represent our state

---

**Qatar**

From Wikipedia, the free encyclopedia

For other places with the same name, see Qatar (disambiguation).

Qatar (الكويك /kəˈtaːr/, /kotar/ or /ɪˈkɔtər/; Arabic: قطر [qatˤar]; local the State of Qatar (Arabic: دولة قطر Dawlāt Qatar), is a sovereign Arab the small Qatar Peninsula on the northeastern coast of the Arabian Penin to the south, with the rest of its territory surrounded by the Persian Gulf. it from the nearby island kingdom of Bahrain. In 2013, Qatar's total populat and 1.5 million expatriates.[8]

---

**Gumbibears**

- Pokemon
- Qatar
- Bahrain

---

[Lyser et al., ACL 2014]
Evaluation methodology

- Mechanical Turk to collect human data
- 7000 questions were answered in the first day
- Over 43000 questions were answered in the space of two weeks
- Total of 461 unique users
- Leaderboard to encourage users

Big problem:

“this man shot at Aaron Burr”

is very different from

“Aaron Burr shot at this man”
Challenge: modeling compositionality

- invented
- double-e
- fridge
- a
- with
- no
- moving

 partes

Gummibears
Pokémon
Qatar
Bahrain

(byyer et al., ACL 2014)
Challenge: modeling compositionality

invented
he fridge
a double-e with parts
no moving
Challenge: modeling compositionality

Algorithms that think on their feet

Hal Daumé III (me@hal3.name)

Iyyer et al., ACL 2014
Challenge: modeling compositionality

\[ e(fridge) = f( w_{fridge} + \mathbb{W} [ e(a) e(double-e) e(with) ]) \]
Results on question-answering task

History Questions

BOW (QB)

RNN (QB)

BOW (wiki)

Combined

History: Model vs. Human

Model loses

Model wins

Score Difference

-200

-150

-100

-50

0

50

100

150

200

25 35 45 55 65 75 85

Full
Results on question-answering task

![Bar chart showing model performance against human in the literature task.

The chart illustrates the score difference between the model and human subjects.

- **Model loses** (red bars)
- **Model wins** (blue bars)

The x-axis represents different score differences, ranging from -400 to 200.

- **Full** (green bars)

The chart compares the performance of different models (BOW, RNN, BOW (wiki), Combined) against the human subjects.

(lyyer et al., ACL 2014)
But the true test...
But the true test... RESULTS!

SUPER thanks to Ken Jennings for being a great sport!
Structured learning with partial feedback

- Loss of a single structured label can be observed
- Labels are never observed
Solution strategy

➢ Use randomization to estimate losses
➢ Apply “standard” learning-to-search to losses
Learning to search

➢ Convert structured prediction into a search problem
  ➢ search space and actions

➢ Define structured features over each state

➢ Construct a reference policy (Ref)
  ➢ Ref usually defined using true label

➢ Learn a policy that imitates Ref
  ➢ Implement with a cost-sensitive classifier
Structured contextual bandit challenge

- True label is not available => Hard to define good Ref
  - Existing L2S algorithms give:
    \[ R(\pi) \leq R(\pi^{\text{ref}}) + o(1) \]
- Can use status quo system as Ref,
  - But competing with this Ref is not useful!

Main goal: Learning to search with:
- A suboptimal reference => improve on Ref
- Partial feedback
Learning to search “schematic”

- Desiridata:
  - Compete with Ref (global opt if Ref is optimal and realizable)
  - Local optimality
## Effect of Roll-{in,out} policies

<table>
<thead>
<tr>
<th>roll-out →</th>
<th>Reference</th>
<th>Mixture</th>
<th>Learned</th>
</tr>
</thead>
<tbody>
<tr>
<td>↓ roll-in</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reference</td>
<td>Inconsistent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learned</td>
<td>No local opt</td>
<td>Good</td>
<td>RL</td>
</tr>
</tbody>
</table>

Mixture: w.p. $\beta$ use Ref, else use Learned.

**Theorem:** LOLS minimizes a combination of regret to Ref and regret to its own one-step deviations.

**Theorem:** Can take $\Omega(2^T)$ steps to reach local optimality.

[Hal Daumé III (me@hal3.name)](mailto:me@hal3.name)
Does it work in practice?

- **Experiments on Dependency Parsing**

<table>
<thead>
<tr>
<th>roll-out → roll-in</th>
<th>Reference</th>
<th>Mixture</th>
<th>Learned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>87.2</td>
<td>89.7</td>
<td>88.2</td>
</tr>
<tr>
<td>Learned</td>
<td>90.7</td>
<td>90.5</td>
<td>86.9</td>
</tr>
</tbody>
</table>

**Reference is optimal**

<table>
<thead>
<tr>
<th>Reference is suboptimal</th>
<th>Reference</th>
<th>Mixture</th>
<th>Learned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>83.3</td>
<td>87.2</td>
<td>81.6</td>
</tr>
<tr>
<td>Learned</td>
<td>87.1</td>
<td>90.2</td>
<td>86.8</td>
</tr>
</tbody>
</table>

**Reference is bad**

<table>
<thead>
<tr>
<th>Reference is bad</th>
<th>Reference</th>
<th>Mixture</th>
<th>Learned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>68.7</td>
<td>65.4</td>
<td>66.7</td>
</tr>
<tr>
<td>Learned</td>
<td>75.8</td>
<td>89.4</td>
<td>87.5</td>
</tr>
</tbody>
</table>

- **LOLS always good, even with Ref is bad**
Learning with partial feedback

- Loss of a *single* structured label can be observed
- Reference policy is *not optimal*

- Apply an $\epsilon$-greedy strategy
- Regret to ref and one-step deviations still bounded
• Reasoning with incomplete information is useful for speed and modeling
• *Imitation learning* can help us build such systems
  • Even when you can't construct a perfect oracle & have incomplete information!
• Wide range of new, interesting problems to work on!
  • *Improve* upon human interpreters?
  • Compete against specific opponents?
  • Distance supervision via structured bandits

Thanks! Questions?